Proceedings of the Third International Conference on Artificial Intelligence and Smart Energy (ICAIS 2023)
IEEE Xplore Part Number: CFP23OAB-ART; ISBN: 978-1-6654-6216-7

2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS) | 978-1-6654-6216-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICAIS56108.2023.10073888

Real Time Object Distance and Dimension Measurement using Deep Learning and OpenCV

Basavaraj M U
Electronics and Telecommunication Engineering
RV College of Engineering®
Bengaluru, Karnataka, India

H Raghuram
Electronics and Telecommunication Engineering
RV College of Engineering®
Bengaluru, Karnataka, India

Mohana
Computer Science & Engineering (Cyber Security)
RV College of Engineering®
Bengaluru, Karnataka, India.

Abstract - **Deep learning is a subset of machine learning based on artificial neural networks in which multiple layers of processing are used to extract progressively higher-level features from data. Deep learning is focused on improving the AI process of having machines learning new things on its own. In this paper, proposed real time object dimension detection and dimension analysis using python. For ML and AI-based papers, python programming language offers consistency, simplicity, and access to excellent libraries and frameworks. It also offers platform freedom, flexibility, and a large developer community. Proposed work, explorations were done on some methodologies, including You Only Look Once (YOLO), for object identification that is meant for speed and real-time application use, and Region-Based Convolutional Neural Networks (R-CNNs) built for model performance and analysis. To make it better, a canny edge detection algorithm is being used. Canny edge detector is a multistage algorithm-based on edge detection operator that can identify a variety of edges in image.**

Keywords -- **OpenCV, Deep Learning, Computer vision, YOLO, R-CNN, Dimension Measurement.**

I. INTRODUCTION

An application called "Real-Time Object Measurement" is used to determine an object's dimensions in real time. In this paper Canny Edge detection algorithm is implemented, AI has been developing rapidly, and there are already a large number of situations in a variety of industries where it is being applied to enhance applications with their work. This is due to AI and its advancements producing accurate results. Proposed method uses a webcam to measure an object's dimension in real-time. A webcam and a white background are required in order to detect an object. Following object detection, it updates an appropriate frame on a screen displaying objects dimension in the desired units in cm. Creating a program that makes use of the OpenCV and NumPy software libraries to implement a suggested method. Requirement for determining each item's size is determining a reference object in order to measure an object's dimension[10] [11]. When dimensions of items inside the reference are measured or computed, measurements are displayed in cms, which is a predetermined unit. At first instance the object is determined and measured in pixels. Now collected data is in pixels which has to be converted into cms.

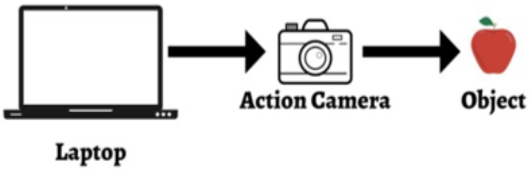


Fig 1. Process of Object dimension measurement

Figure 1 shows the process of object dimension measurement this method is the best and most convenient method to measure the dimension of an object for various computer vision applications with good future scope[14] [15].

II. LITERATURE SURVEY

Zaarane A, *et al* [1] discusses inter-vehicles distance measurement, this method used for many real time applications regardless of object types. Arturo F *et al* [2] presents a method measuring the distance between a camera and a human head in two dimensional images using a camera. Also Leading head pose estimation algorithms are based mainly on head orientation and translations normal to the camera's orientation. Clément Godard, *et al* [3] focuses on improving current methods and replacing the usage of explicit depth data during training to make getting binocular stereo film easier. Adrian Rosebrock *et al* [4] discussed how OpenCV in Python can be used to calculate the distance with a webcam between two points. One of the applications is in image and video processing to detect objects, faces, or handwriting of a human. It is also used in finding objects. G V Shalini *et al* [5] presents a method for ensuring social distancing between people in order to curb the spread of COVID-19. It takes reference from video frames to find out a standard or safe gap between individuals to reduce spreading of pandemic. Results and outcomes can be used in public areas to control the spreading of viruses. Chen W *et al* [6] proposes a unique algorithm called depth-assisted edge detection and tries to improve algorithm i.e., depth map inpainting by taking extracted edges. Color image and raw depth data are very important parameters in obtaining initial edges. Eventually optimizing edge orientation helps in assisting depth map inpainting. V S K P Varma *et al* [7] in this paper proposes a method which is helpful in finding approaching hump/bump and also in visualizing speed at which incoming hump/bump approaches a driver by utilizing DL techniques and its o/p is how far from vehicle is away from incoming hump/bump by techniques such as stereo-vision approaches.

III. DESIGN AND IMPLEMENTATION

OpenCV-A computer vision and machine learning software library called OpenCV is available for free which uses a library of programming functions called OpenCV that is primarily focused on real-time computer vision. The purpose of OpenCV was to create a standard infrastructure for computer vision applications and to make it easier for businesses to utilize artificial intelligence in their products.
NumPy- Numerical Python is referred to as NumPy. It has excellent mathematical capabilities. Working with arrays requires usage of a Python package called NumPy.
Imutils- Imutils is a set of OpenCV functions that simplifies fundamental image processing operations like image translation, rotation, and resizing.

A. *Block diagram*

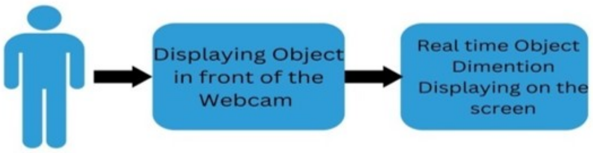

Displaying Object in front of the Webcam
Real time Object Dimention Displaying on the screen

Fig 2. Block Diagram of object dimension measurement

Figure 2 shows the block diagram of object dimension measurement. It gives a detailed view and working of real time object dimension measurement. Object is captured by the camera and object is detected [9]. further the coordinates are established around the object and these coordinates are joined to form the frame that encloses the object, resulting in determining dimensions of the object .

B. *Flowchart*

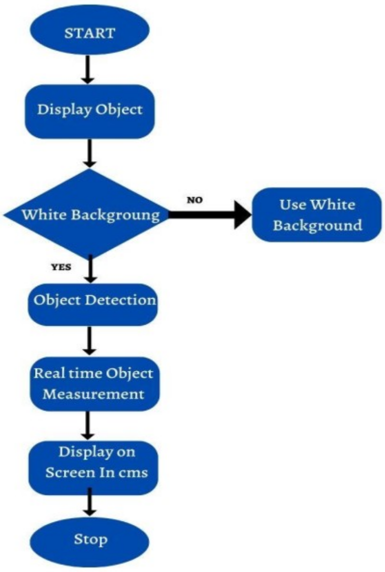
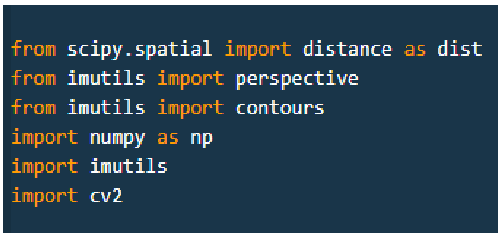


Fig 3. Model Design and flow chart of object dimension measurement

Figure 3 shows the model design and flow chart of object dimension measurement. It has four steps such as capturing an image, object measurement process, saving the output and displaying output. Download the dependencies OpenCV and NumPy in Spyder. Import OpenCV as cv2 and NumPy as np.

Capturing an Image



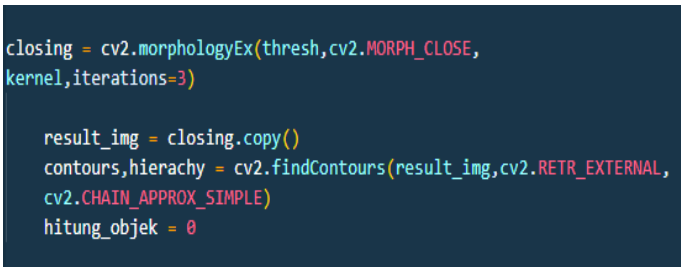
```
from scipy.spatial import distance as dist
from imutils import perspective
from imutils import contours
import numpy as np
import imutils
import cv2
```

Fig 4.Installing important libraries

After importing cv2 and NumPy follow a suit and install imutils. First, fix the camera and sensor in a linear fashion. This is an initial part after defining a midpoint as function Spyder(anaconda3). Now to capture an image *cap = cv2.VideoCapture(0)* syntax is used. This syntax uses *VideoCapture(0)* as a member function of cv2, which helps in excitation. Start a camera and procedure of capturing an image using class cap and read functions are sent in a loop with required parameters that have to be met. Under greyscaling various properties of objects such as color, pixels, thresh and kernel are declared and defined. further the capturing process is initialized using a function[12] [13].

Formation of Frame-Imutils is used to carry out fundamental image processing operations like skeletonization, contour sorting, and edge detection [8]. The library is used to import contours. NumPy np, arg parse is also imported in a similar fashion. The object is then surrounded by co-ordinates, and a midpoint(pt A, pt B)function that takes two arguments, pt A and pt B, and returns the object's midpoint from the reference point, is then used. The object's boundaries are drawn with the desired color and width using another similar procedure. This causes a rectangular box to form around the object.

Object Measurement Process

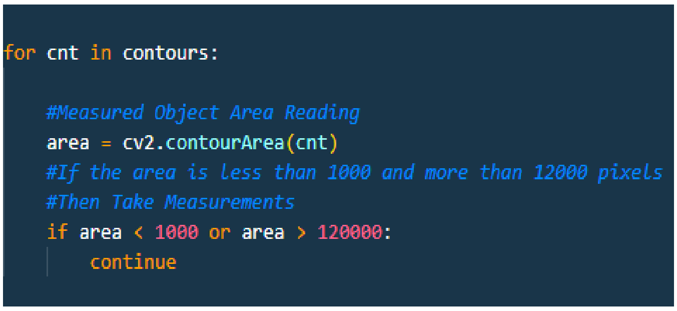


```
closing = cv2.morphologyEx(thresh,cv2.MORPH_CLOSE,
kernel,iterations=3)

    result_img = closing.copy()
    contours,hierachy = cv2.findContours(result_img,cv2.RETR_EXTERNAL,
    cv2.CHAIN_APPROX_SIMPLE)
    hitung_objek = 0
```

Fig 5. Initialization and declaring the code for capturing an image

Initialization and declaring the code for capturing an image step as shown in figure 5. Next step i.e., measurement of an object, contours are used which is a python list of all contours of an image. Contours which are defined as lines joining all points along edges of an image consisting of the same shaping analysis and object detection. Performs different functions such as area, perimeter etc. A NumPy array of arbitrary (x,y) coordinates of an object's boundary makes up each individual contour. One such function used is Area() which is called using cv2 class with certain constraints.

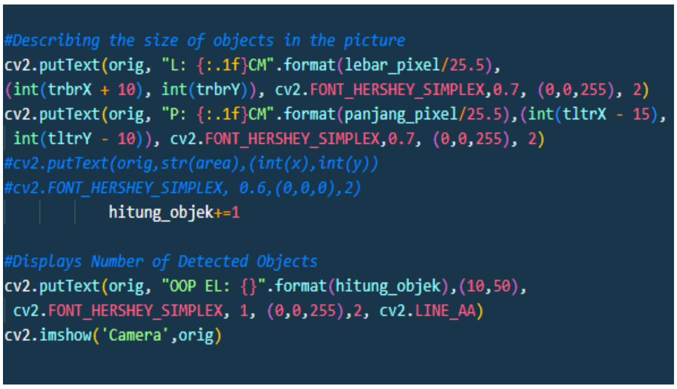


```
for cnt in contours:

    #Measured Object Area Reading
    area = cv2.contourArea(cnt)
    #If the area is less than 1000 and more than 12000 pixels
    #Then Take Measurements
    if area < 1000 or area > 120000:
        continue
```

Fig 6. Determining area of object in real time

Figure 6 shows the steps of object area measurement in real time. Before saving data from an input a few parameters have to be fetched. Program has to draw boxes around required objects using minArea() and imutils of a program. Then circles are drawn using cv2.circle() which takes valid arguments as image, center coordinates, radius, color, thickness of the boss and edge. After all this process we have to decide the font size, color and style. Before doing that, pixels, units of input data have to be converted into desired dimensional units for example cms, inches, meters etc. After built-in function cv2 is used to determine required font-related parameters. Units like cm, inch, and meter measurement units are displayed using a putText() function, and their size can be adjusted using pixels and perspective transformation. So now to display an image on a display or a screen it uses for example laptop camera with the help of reference function.



```
#Describing the size of objects in the picture
cv2.putText(orig, "L: {:.1f}CM".format(lebar_pixel/25.5),
(int(trbrX + 10), int(trbrY)), cv2.FONT_HERSHEY_SIMPLEX,0.7, (0,0,255), 2)
cv2.putText(orig, "P: {:.1f}CM".format(panjang_pixel/25.5),(int(tltrX - 15),
 int(tltrY - 10)), cv2.FONT_HERSHEY_SIMPLEX,0.7, (0,0,255), 2)
#cv2.putText(orig,str(area),(int(x),int(y))
#cv2.FONT_HERSHEY_SIMPLEX, 0.6,(0,0,0),2)
            hitung_objek+=1

#Displays Number of Detected Objects
cv2.putText(orig, "OOP EL: {}".format(hitung_objek),(10,50),
 cv2.FONT_HERSHEY_SIMPLEX, 1, (0,0,255),2, cv2.LINE_AA)
cv2.imshow('Camera',orig)
```

Fig 7. Initializing and determining font size of the border and display

The above function as mentioned earlier is a reference function in which necessary changes are made in the original will change in a reference object and changes in a reference copy will change input data in the original. Function defined will also give a number of objects detected and displayed. In this paper, Canny Edge detection algorithm is implemented. It follows a multistage algorithm to identify items of a wide range. It includes Noise reduction - Edge detection being vulnerable to image noise, and it is implemented by edge tracking by Hysteresis. At this stage, classification of real edges is performed. So, to classify between sharp and curved edges two values maxVal and minVal are taken as reference values. Any value above maxVal classifies it as a sharp edge and any value below minVal classifies it as curve edges. So, the values between maxVal and minVal are filtered using sure-edge filter criteria. To locate contours and apply procedures like code, to an input image so as to extract the real-time object dimension, a python file (utlis.py) is written.

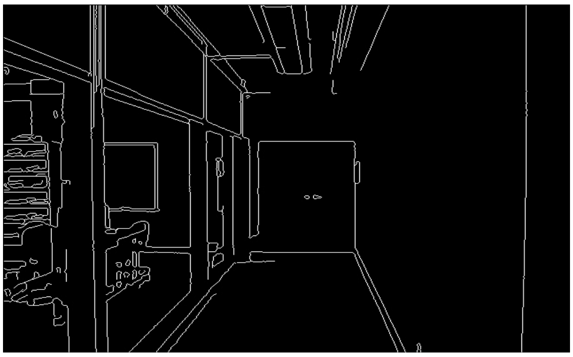


Fig 8.Canny image after dilation and erosion operation

Figure 8 shows the canny image after dilation and erosion operation. The image is transformed into a clever image. By exploiting dilation and erosion properties, a smooth canny image creation process can be achieved. Writing code to identify background white paper's limitation, which is contained within a rectangle frame. Applying passing through filters to a white paper background as a rectangle. Adding length, area and finishing curves of an object as contours. Consequently, defining background screen detection. Measure an object that is on a white paper with help of an aid of a canny image by using mathematical principles and a clever illustration. Now a data or text is to be displayed with an estimated measurement of displayed object in cms.

IV. SIMULATION RESULTS AND ANALYSIS

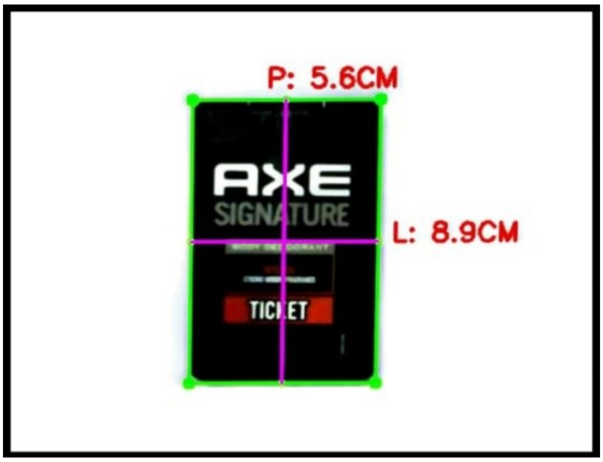


Fig 9.Measuring the object 1 Dimension.

Figure 9 shows the object 1 dimension measurement. Measured length and width is *Length: 8.9cm* and *Width: 5.6cm.* By utilizing Python's OpenCV and measured distance from camera. Firstly, need to create an environment for a paper by installing prerequisites modules in spyder (integrated development environment). Now, a webcam is connected to a laptop used in an experiment for measuring an object's dimension. In continuation, objects are now detected, and coordinates of displayed objects are determined. Contour present inside an OpenCV library will produce a perimeter and area by joining coordinate points as shown in Figure 10.

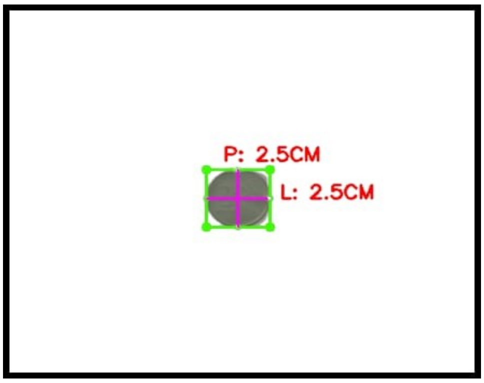


Fig 10. Determining the object dimension(coin).

In this paper, a rectangular object's dimension is determined by a computer. Instead of using coins, because coins are circular in shape, a phone is used to verify model working. However, the proposed model will identify their dimensions using rectangular dimension parameters.

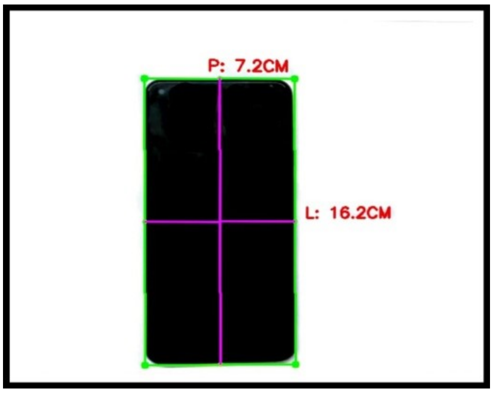


Fig 11. The dimension measurement of mobile.

Figure 11 shows the dimension measurement of mobile. Measured dimensions are *Width of the object w = 7.2cm Length of the object l = 16cm.*

Inference and performance Analysis

TABLE 1 OBJECT DIMENSION MEASUREMENT

Practical Measured Value			Error at different value of d			
Width(w)	Length (l)	Area (A)	Ew(%)	El(%)	EA(%)	Distance (d)
8.6	18.8	161.68	19.44	17.5	40.35	Min=27
8.5	18.2	154.7	18.06	13.75	34.3	27.4
7.2	16.1	115.92	0	0.625	0.625	31.2
5.8	13	25.4	19.44	18.75	34.55	36
4.8	10.3	51.29	33.33	35.625	55.5	46
2.6	5.8	15.08	63.88	63.75	86.909	82.5

Table I represents the relation between length, breadth, width and the area of objects at various distances and provides the corresponding values, measured from the webcam. The camera of a laptop which is used in this experiment and the object are separated by distance(d). The practical measured values of the object are taken at different values of distance(d) from the laptop. Define 3 different points basically from the above table. Critical Point(dcp) b) Accurate Point(dap) c) Out of range point(dop)

Critical point- Minimum distance between camera of laptop and the object required below which accurate measurement of the object's dimensions is impossible. Symbol is dcp where dcp = 27 cm.

Accurate point:- The distance between the subject and the laptop's camera at which the area and dimension of the theoretical and actual values are virtually identical, or at which the error is at its smallest, is indicated by the symbol dap. Here, dap = 27.4 cm.

Out of range point:- Maximum distance of subject from the camera of a laptop above which error in length and width will be almost equal and error in area will be almost 100%. It is denoted by dop. Here dop = 75 cm and above Length angle(x), Width angle(y) and Area angle(z):

Length angle(x): It is defined as $x= atan(l*d/2)$

Width angle(y): It is defined as $y= atan(w*d/2)$

Area angle(z): It is defined as $z= atan(Ap*At/2)$

Ap:Area of the object measured at different values of d.

At: Area of the object measured theoretically, At=115.2sq cm

V. CONCLUSION

This paper presents an enhanced technique for a real-time measurement of the object's dimensions. Several functions in OpenCV can be used to determine an object's length, breadth, and volume. Drawing a box all the way around an object is the simplest technique to measure its dimensions. Next, utilize the associated OpenCV functions to determine the box's length, breadth, and height. These measurements can then be applied to other things. Therefore, the computer vision (webcam device and code) is employed to measure the object dimensions in real time in cm. It takes the image from the frame of the real-time video and then papers its dimensions on the screen. It is successful to determine the object's dimensions using a Canny edge detector. This method is quick to use, has a lot of benefits, and notable qualities that can be used in the real world.

REFERENCES

[1] Zaarane A, *et al.* “Distance measurement system for autonomous vehicles using stereo cameras”. 2020.

[2] Arturo F, *et al.* “Camera distance from face images”.2013.

[3] C. Godard *et al.* “Unsupervised Monocular Depth Estimation with Left-Right Consistency, CVPR”, 2017.

[4] Adrian Rosebrock *et al.* “Find distance from camera to object/marker using Python and OpenCV”, 2015.

[5] G V Shalini *et al.* “Social Distancing Analyzer Using Computer Vision and Deep Learning”, 2021.

[6] Chen.W, *et al.* “An improved edge detection algorithm for depth maps in painting”, *Optics and Lasers in Engineering, 2014*, pp.69–77.

[7] V S K P Varma, *et al* “Real Time Detection of Speed Hump/Bump and Distance Estimation with Deep Learning using GPU and ZED Stereo Camera”, 2018.

[8] Muthu Krishnan R *et al.* “Edge Detection Techniques for Picture Segmentation” 2011.

[9] Mengel, P *et al.* “camera necessities and particulars” Client Cams paper deliverable. 2013.

[10] Robert J W *et al.* “A real-time object detection system on mobile devices”. 2018.

[11] Joseph R, *et al.* “YOLO: Unified, real-time object detection”.2016.

[12] S. K. Mankani, *et al.* “Real-time implementation of object detection and tracking on DSP for video surveillance applications,” *International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT),* 2016, pp. 1965-1969.

[13] A. Biswas *et al.* “Classification of Objects in Video Records using Neural Network Framework,” *International Conference on Smart Systems and Inventive Technology (ICSSIT),* 2018, pp. 564-569.

[14] E. Shreyas et al. “3D Object Detection and Tracking Methods using Deep Learning for Computer Vision Applications,” *International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT),* 2021, pp. 735-738.

[15] V. Rajesh *et al.* "Quantum Convolutional Neural Networks (QCNN) Using Deep Learning for Computer Vision Applications,” *International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT),* 2021, pp. 728-734.